

Control 7 segment digits by keyboard

In the previous tutorial, we learn about uart data transmit. In this tutorial, we will start uart receive data from the PC terminal to the STM32 dev board and Control 7-segment display digit by simply typing the keyboard digit.

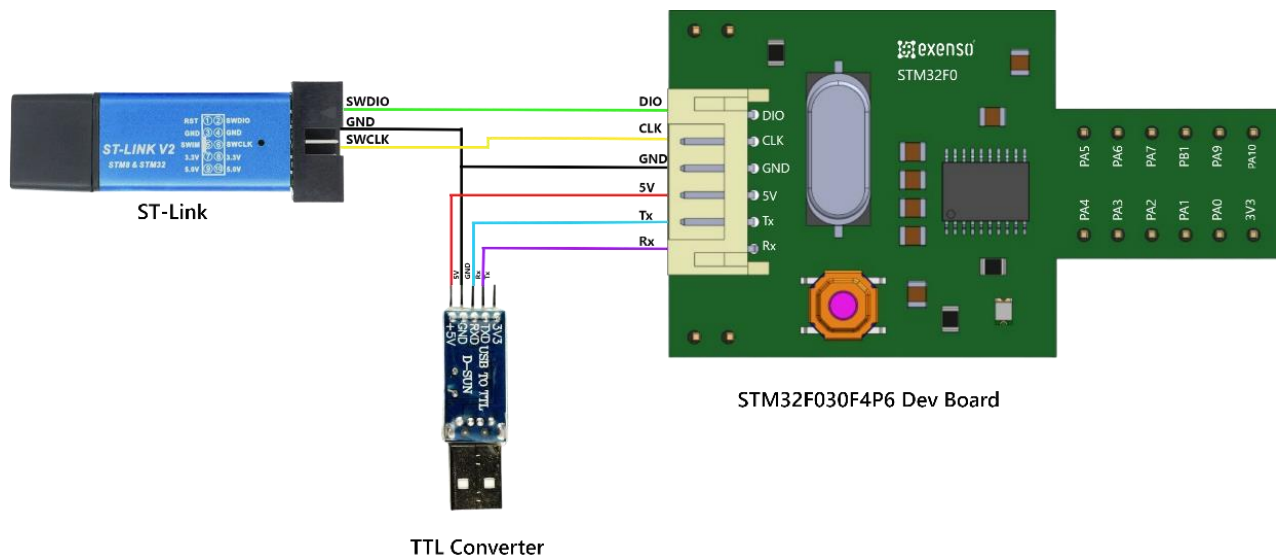
Components Required

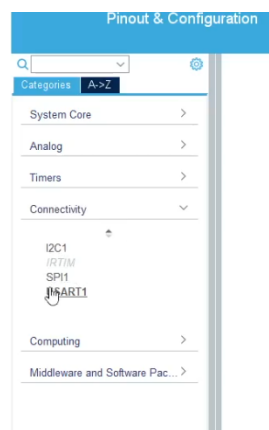
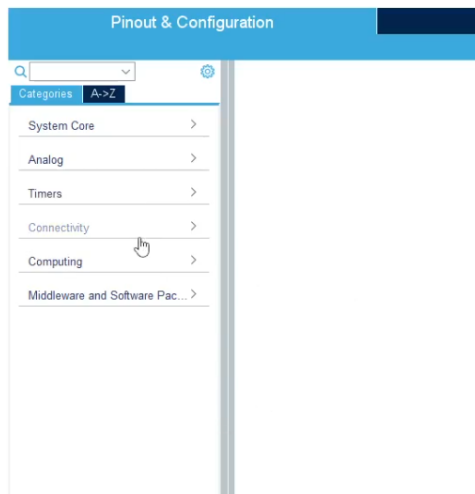
You will need the following components –

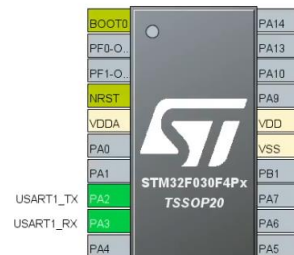
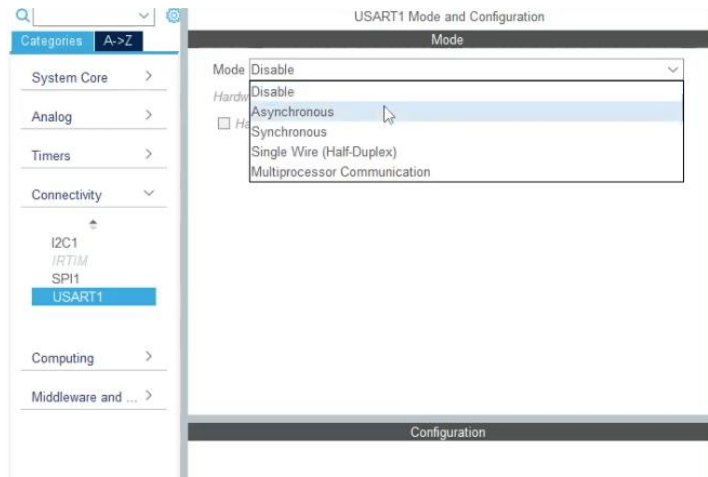
- 1 × Breadboard
- 1 × STM32F030F4P6
- 1× TTL Converter
- 1× Single digit Seven Segment display
- 8 × 100Ω Resistor
- Some Jumper wire

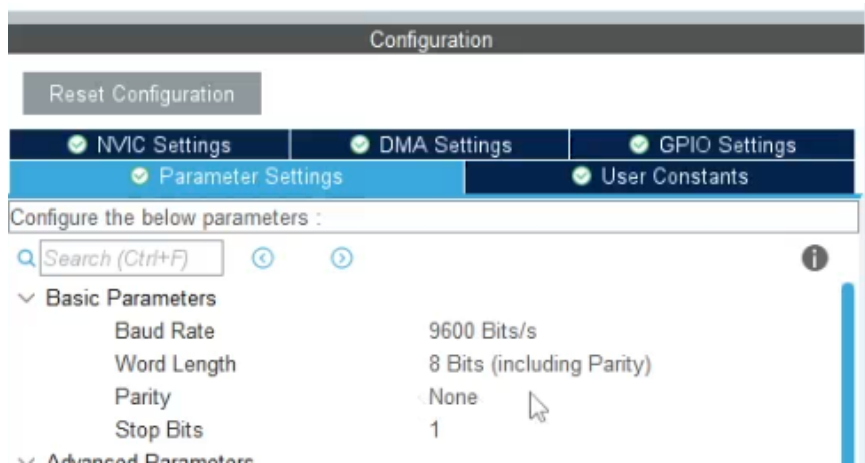
Procedure

Follow the circuit diagram shown in the image given below.

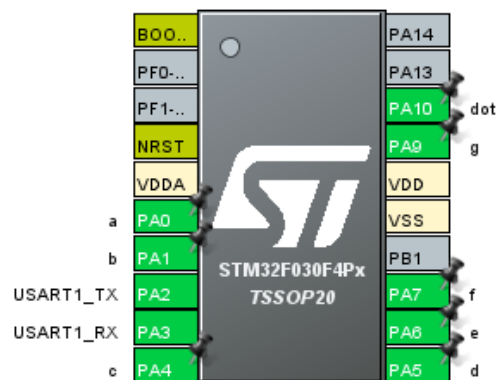








PA0 to PA1 and PA4 to PA10 set as OUTPUT



Code

```
#include "main.h"

UART_HandleTypeDef huart1;
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_USART1_UART_Init(void);

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_USART1_UART_Init();
    uint8_t UART1_rxBuffer[2] = {0};
    while (1)
    {
        HAL_UART_Receive(&huart1, UART1_rxBuffer, 2, 100);

        switch (UART1_rxBuffer[0]){
            case '0':
                // statements
                //0
                HAL_GPIO_WritePin(GPIOA, a_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOA, b_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOA, c_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOA, d_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOA, e_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOA, f_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOA, g_Pin, GPIO_PIN_SET);
                HAL_GPIO_WritePin(GPIOA, dot_Pin, GPIO_PIN_SET);

                break;

            case '1':
                // statements
                //1
                HAL_GPIO_WritePin(GPIOA, a_Pin, GPIO_PIN_SET);
                HAL_GPIO_WritePin(GPIOA, b_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOA, c_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOA, d_Pin, GPIO_PIN_SET);
                HAL_GPIO_WritePin(GPIOA, e_Pin, GPIO_PIN_SET);
                HAL_GPIO_WritePin(GPIOA, f_Pin, GPIO_PIN_SET);
                HAL_GPIO_WritePin(GPIOA, g_Pin, GPIO_PIN_SET);
                HAL_GPIO_WritePin(GPIOA, dot_Pin, GPIO_PIN_SET);

                break;

            case '2':
                // statements
                //2
                HAL_GPIO_WritePin(GPIOA, a_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOA, b_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOA, c_Pin, GPIO_PIN_SET);
                HAL_GPIO_WritePin(GPIOA, d_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOA, e_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOA, f_Pin, GPIO_PIN_SET);
                HAL_GPIO_WritePin(GPIOA, g_Pin, GPIO_PIN_RESET);
                HAL_GPIO_WritePin(GPIOA, dot_Pin, GPIO_PIN_SET);

                break;
        }
    }
}
```

```

case '3':
    // statements
    //3
    HAL_GPIO_WritePin(GPIOA, a_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, b_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, d_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, e_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, f_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, g_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, dot_Pin, GPIO_PIN_SET);

    break;
case '4':
    // statements
    //4
    HAL_GPIO_WritePin(GPIOA, a_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, b_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, d_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, e_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, f_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, g_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, dot_Pin, GPIO_PIN_SET);

    break;
case '5':
    // statements
    //5
    HAL_GPIO_WritePin(GPIOA, a_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, b_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, d_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, e_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, f_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, g_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, dot_Pin, GPIO_PIN_SET);

    break;
case '6':
    // statements
    //6
    HAL_GPIO_WritePin(GPIOA, a_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, b_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, d_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, e_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, f_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, g_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, dot_Pin, GPIO_PIN_SET);

    break;
case '7':
    // statements
    //7
    HAL_GPIO_WritePin(GPIOA, a_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, b_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOA, d_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, e_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, f_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, g_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, dot_Pin, GPIO_PIN_SET);

    break;

```

```

    case '8':
        // statements
        //8
        HAL_GPIO_WritePin(GPIOA, a_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, b_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, c_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, d_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, e_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, f_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, g_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, dot_Pin, GPIO_PIN_SET);

        break;
    case '9':
        // statements
        //9
        HAL_GPIO_WritePin(GPIOA, a_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, b_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, c_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, d_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, e_Pin, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, f_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, g_Pin, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, dot_Pin, GPIO_PIN_SET);

        break;

    default:

        HAL_GPIO_WritePin(GPIOA, a_Pin, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, b_Pin, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, c_Pin, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, d_Pin, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, e_Pin, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, f_Pin, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, g_Pin, GPIO_PIN_SET);

        // default statements
}

}

}

```

```

HAL_UART_Receive(UART_HandleTypeDef *huart, uint8_t *pData, uint16_t Size, uint32_t
Timeout)

```